



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/052,250

01/23/2002

Russell T. Davis

7643.0042

1920

22852

7590

11/24/2008

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP

901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER

NGUYEN, CHAU T

ART UNIT

PAPER NUMBER

2176

MAIL DATE

DELIVERY MODE

11/24/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/052,250
Filing Date: January 23, 2002
Appellant(s): DAVIS ET AL.

Jeffrey A. Berkowitz
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/28/2008 appealing from the Office action mailed 11/01/2007.

(1) Real Party in Interest

The Appellant's statement of the real party in interest contained in the brief is correct.

(2) Related Appeals and Interferences

The Examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The Appellant's statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments

The Appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of The Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The Appellant's statement on the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

The following is a listing of the evidence relied upon in the rejection of claims under appeal:

Krug et al., US Patent No. 6,721,736, issued on April 13, 2004, but filed on November 15, 2000 (hereinafter Krug).

Walter Hamscher, "Extensible Business Reporting Language (XBRL) Specification, published on July 31, 2000.

Saxton, US Patent No. 6,370,549, issued on April 9, 2002, but filed on January 4, 1999.

Polizzi et al., US Patent Application Publication No. US 2002/0052954 A1, Published on May 2, 2002, but filed on April 27, 2001.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 62-64 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krug et al. (Krug), US Patent No. 6,721,736 and further in view of further in view of Hamscher et al. (Hamscher), published on 07/31/2000.

As to claim 62, Krug discloses a data processing comprising:

a parser that (col. 8, lines 21-30: a syntax tree parser):

receives one or more text documents (col. 7, lines 46-50, col. 8, lines 21-30, and Fig. 3: HTML document is inputted to the interface 8)

interprets tags included in the one or more text documents to create software elements (col. 8, lines 21-30: the syntax tree parser analyses the HTML document by recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements), and

determines the hierarchy of the software elements within a structure representative of the one or more text documents (col. 8, lines 21-30: recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements).

However, Krug does not explicitly disclose a manager that provides for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.

Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances, and taxonomies can be composed together to extend other taxonomies (page 1, Abstract

Art Unit: 2176

and page 6). Hamscher also discloses combining of financial information from different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include providing for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document. Hamscher suggests that XBRL allows software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

As to claim 63, Krug discloses the structure is a Numerator Document Object Model (NDOM) (col. 7, lines 46-50, col. 8, lines 21-30, and Fig. 3).

As to claim 64, Krug disclose a parser that receives text documents, interprets tags of the text document, and determines the hierarchy of the software elements within the text document as discussed in claims 62-63 above.

However, Krug does not explicitly disclose wherein the one or more text documents are XBRL documents.

Hamscher discloses XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances (pages 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

Claims 1-6, 11-21, 24-33, 34, 37-46, 49-57, and 59-61 are rejected under 35 U.S.C. 103(a) as being unpatentable over Saxton, US Patent No. 6,370,549 in view of Polizzi et al. (Polizzi), US Patent Application Publication No. 2002/0052954 and further in view of Hamscher et al. (Hamscher), published on 07/31/2000.

As to claims 1, 17, 29, 30, 42, and 54, Saxton discloses a data processing system for developing reports, comprising:

a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents (Abstract, col. 5, lines 63 – col. 6, lines 16);
and

However, Saxton does not explicitly disclose an editor that develops a report by referencing the software elements created from the one or more text documents to form

Art Unit: 2176

a structure of the report and retrieves data from one or more sources to represent one or more values within the report and wherein a mapper generates a relationship between the data from the one or more sources and the one or more values to be placed within the report.

Polizzi discloses in Abstract, page 3, paragraph [0024] and page 6, paragraph [0039]: a repository stores all computer files, which are called objects, and the objects can be any computer file such as text documents; these text documents are organized or arranged in a hierarchy). Polizzi also discloses preparing a report based upon retrieved data (Abstract), thus a mapper must be inherent from Polizzi's system since it prepared a reports based upon retrieved data (a relationship between the data from the one or more sources and the one or more values to be placed within the report).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include develops reports by referencing the software elements created from the one or more text documents and retrieves data from one or more sources to represent one or more values within the report and wherein a mapper generates a relationship between the data from the one or more sources and the one or more values to be placed within the report in order to allow user to view and select reports.

However, Saxton and Polizzi do not explicitly disclose a manager that provides for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.

Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances, and taxonomies can be composed together to extend other taxonomies (page 1, Abstract and page 6). Hamscher also discloses combining of financial information from different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher with Saxton and Polizzi to include providing for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document. Hamscher suggests that XBRL allows software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

As to claims 2, 18, 31, 43, and 55, Saxton and Polizzi (Saxton-Polizzi) disclose wherein the format with the hierarchal relationship between the software elements is a Numerator Document Object Model (NDOM) (Saxton, Abstract).

As to claims 3, 19, 32, 44, and 56, Saxton-Polizzi, however, do not disclose wherein the one or more text documents are XBRL documents.

Hamscher discloses XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances (pages 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Saxton-Polizzi to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

As to claims 4, 20, 33, 45, and 57, Saxton-Polizzi, however, do not disclose wherein the parser creates the software elements having the format with the hierarchal relationship by interpreting tags included in the one or more text documents.

Hamscher discloses in pages 1-3 and 7 that XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances (pages 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Saxton-Polizzi to include XBRL documents because XBRL documents allow software vendors,

Art Unit: 2176

programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

As to claim 5, Saxton-Polizzi disclose wherein a manager manipulates the software elements (Saxton, col. 5, line 63 – col. 6, line 65).

As to claims 6, 21, 34, and 46, Saxton-Polizzi disclose wherein the manager manipulates the software elements by browsing, editing, loading, and storing the software elements (Saxton, col. 6, line 66 – col. 7, line 11; Polizzi, Abstract, and page 1, paragraphs [0005]-[0006]).

As to claim 11, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not disclose wherein a mapper links the report and the one or more sources that will present one or more values within the report.

Polizzi discloses in the Abstract, pages 3-4, paragraphs [0024]-[0026], and page 6, paragraph [0039]: a repository stores all computer files, which are called objects, and the objects can be any computer file such as text documents; these text documents are organized or arranged in a hierarchy). Polizzi also discloses preparing a report based upon retrieved data (Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include develops reports by referencing the software elements created from the one or more text documents and retrieves data from one or more sources to represent one or more values within the report in order to allow user to view and select reports.

As to claims 12, 24, 37 and 49, Saxton-Polizzi disclose wherein the report and the one or more sources are linked through a “drag and drop” process (Saxton, col. 1, line 61 – col. 2, line 12 and col. 7, lines 12-24).

As to claims 13, 25, 38, 50, and 59, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not explicitly disclose wherein the editor provides for the software elements to be modified to create a new combination of software elements representative of a new text document.

Polizzi discloses object or category within the repository is modified (page 4, paragraph [0027]).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include modified

Art Unit: 2176

object or category in order to provide a copy of the output report as an automatic update to a user's page.

As to claims 14, 26, 39, 51, and 60, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not explicitly disclose wherein the editor provides for modification of one or more parameters associated with the software elements.

Polizzi discloses object or category within the repository is modified (page 4, paragraph [0027]).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include modified object or category in order to provide a copy of the output report as an automatic update to a user's page.

As to claims 15, 27, 40, 52 and 61, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not explicitly disclose wherein the software elements are transformed to new software elements and are imported into an RDL system.

Polizzi discloses in the Abstract, page 2, paragraphs [0008] and [0020]: the portal page is an object arranged in a format that is readable by a browser program, and the user interface may be based upon a standard browser program that is capable of reading HTML, Java, XML, or other languages.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include using a standardized browser program as a user interface so the user can point and click on hypertext links to navigate through the portal system which provides the ability to search both structured and unstructured data.

As to claims 16, 28, 41, and 53, Saxton-Polizzi disclose wherein the software elements are transformed to the new software elements by retrieving a tag associated with each of the software elements in a dictionary and invoking a translation routine associated with the tag (Saxton, col. 7, line 44 – col. 10, line 30).

Claims 8-10, 23, 36, 47-48, and 58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Saxton, Polizzi et al. (Polizzi) and Hamscher as applied to claims 1-6, 11-21, 24-33, 34, 37-46, 49-57, and 59-61 above, and further in view of Clancey et al. (Clancey), US Patent No. 6,134,563.

As to claim 8 and 47, Saxton-Polizzi-Hamscher, however, do not explicitly disclose wherein one or more templates are used to develop the report, which contain

Art Unit: 2176

data that is directly inserted into the report and instructions enabling data from the one or more source to be inserted into the report.

Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Clancey and Saxton-Polizzi-Hamscher to include using templates to develop the report which contain data that is directly inserted into the report and instructions enabling data from the one or more source to be inserted into the report. By using templates to develop reports, it would provide user-friendly environment and save time for users.

As to claims 9, 23, 36, and 58, Saxton-Polizzi-Hamscher, however, do not explicitly disclose wherein the one or more templates contain data that is directly inserted into the report and instructions enabling data from the one or more sources to be inserted into the report.

Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Clancey and Saxton-Polizzi-Hamscher to include using templates to develop the report. By using templates to develop reports, it would provide user-friendly environment and save time for users.

As to claims 10 and 48, Saxton-Polizzi-Hamscher, however, do not explicitly disclose wherein the one or more templates provide instructions to a mapper to retrieve the data that is directly inserted into the report and data from local or remote sources.

Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Clancey and Saxton-Polizzi-Hamscher to include using templates to develop the report. By using templates to develop reports, it would provide user-friendly environment and save time for users.

(10) Response to Argument

In the Remarks, Appellant argued in substance that

A. The rejection of claims 62-64 under U.S.C. § 103 as being unpatentable over Krug and Hamscher is improper

A-1. “Appellants submit that a prima facie case of obviousness has not been established.” (see page 15 of Brief)

In reply to argument A-1, to establish a prima facie case of obviousness, three basic criteria must be met.

First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. In this case, Krug discloses recognizing the HTML tags within the document and constructing a hierarchical HTML

Art Unit: 2176

syntax tree that represents the hierarchical relationship of the syntax elements (software elements), which is similar to XBRL consisting of XML elements and attributes of Hamscher, thus it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

Second, there must be a reasonable expectation of success. The prior art can be modified or combined to reject claims as prima facie obvious as long as there is a reasonable expectation of success. In *re Merck & Co., Inc.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). In this case, claimed invention directed to a method for data processing comprising a parser for receiving text documents which are XBRL documents, interpreting tags of the documents and determining the hierarchy of the software elements of the documents, was rejected as obvious over a reference (Krug) which taught recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements) and further in view of Hamscher reference which taught XBRL consisting of XML elements and attributes. Thus, there was reasonable expectation that a process combining the prior art steps could be successfully scaled up.

Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. In this case, Krug discloses a data processing comprising:

a parser that (col. 8, lines 21-30: a syntax tree parser):

receives one or more text documents (col. 7, lines 46-50, col. 8, lines 21-30, and Fig. 3: HTML document is inputted to the interface 8)

interprets tags included in the one or more text documents to create software elements (col. 8, lines 21-30: the syntax tree parser analyses the HTML document by recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements), and

determines the hierarchy of the software elements within a structure representative of the one or more text documents (col. 8, lines 21-30: recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements).

However, Krug does not explicitly disclose a manager that provides for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.

Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances, and

Art Unit: 2176

taxonomies can be composed together to extend other taxonomies (page 1, Abstract and page 6). Hamscher also discloses combining of financial information from different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include providing for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document. Hamscher suggests that XBRL allows software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

A-2. The syntax elements of Krug cannot constitute the claimed "software elements," which are created by interpreting "tags" in the "text documents." (see page 16 of Brief)

In reply to argument A-2, claim 62 claimed "interprets tags included in one or more text documents to create software elements, and determines the hierarchy of the software elements within a structure representative of the one or more text document". Krug discloses analyzing the HTML syntax structure of the search result document by recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software

Art Unit: 2176

elements) (col. 8, lines 20-30). Thus, the HTML document is transformed into a syntax tree representing the hierarchical relationship of the syntax elements.

A-3. Hamscher does not teach or suggest the claimed "provides for the creation of a second hierarchy between the software elements, and provides for the restructuring of the first hierarchy and second hierarchy into software structures corresponding to a new text document." (see page 18 of Brief)

In reply to argument A-3, Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and *taxonomies of elements (software elements) referred to in document instances*, and *taxonomies can be composed together to extend other taxonomies* (page 1, Abstract and page 6). Thus, document instances (hierarchies) can be composed together to extend to other hierarchies, and one of ordinary skill in the art would have interpreted that as if one document instance (hierarchy) is created and extended to other hierarchies (second hierarchy). Page 5 of Hamscher's reference discloses that *taxonomy is an XML schema that defines new elements each corresponding to a concept that can be referenced in XBRL documents*. Hamscher also discloses combining of financial information from different periods or entities or even for the same entity under different reporting regimes, and *creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents* (page 17, 1st paragraph).

Art Unit: 2176

B. The rejection of claims 1-6, 11-21, 24-34, 37-46, 49-57, and 59-61 under 35 U.S.C. § 103(a) as being unpatentable Saxton Polizzi, and Hamscher is improper.

B-1. Hamscher does not teach the claimed “creation of a second hierarchy of the software elements.” (see page 19 of Brief)

In reply to argument B-1, this argument is similar to argument A-3, which is already addressed in argument A-3. Please see the reply to argument A-3 above.

C. The rejection of claims 8-10, 23, 36, 47, 48, and 58 under 35 U.S.C. § 103(a) as being unpatentable over Saxton, Polizzi, Hamscher, and Clancey is improper.

C-1. Clancey fails to cure the deficiencies of Saxton, Polizzi, and Hamscher and Clancey does not teach or suggest “a manager that provides for the creation of a second hierarchy between the software elements and the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document,” as recited in claim 1. (see page 21 of Brief)

In reply to argument C-1, Examiner has used Clancey reference to reject the limitation “one or more templates are used to develop the report, which contain data that is directly inserted into the report and instructions enabling data from the one or more source to be inserted into the report” **NOT** “a manager that provides for the creation of a second hierarchy between the software elements and the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.” Therefore, the Appellant cannot argue that Clancey reference does not

Art Unit: 2176

teach "a manager that provides for the creation of a second hierarchy between the software elements and the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document."

In this case, Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

Art Unit: 2176

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is respectfully submitted that the rejections should be sustained.

Respectfully Submitted,

/Chau Nguyen/
Chau Nguyen

Conferees:

/Rachna S Desai/
Primary Examiner, Art Unit 2176

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100